Web4: A Seven-Layer Sovereign-Ownership Architecture for the Next Generation of the Internet

The Ownership Thesis, Layered Reference Architecture, and Security Analysis

Marc Herdina
LINKSPREED GROUP — Web4 Division
hello@linkspreed.com

ABSTRACT

Abstract*—The evolution of the World Wide Web is conventionally described in three eras — Web1 (read), Web2 (read-write), and Web3 (read-write-own) — each of which relocated a distinct form of power along the value chain: from publishers, to platforms, to ledger-anchored token holders. This paper introduces and defends a fourth: Web4 (read-write-own-act), in which sovereign ownership is not confined to tokens or ledger records but is co-specified across every operational plane of the network — identity, physical connectivity, compute, naming, developer tooling, application delivery, and consumer hardware — and in which autonomous agents act on behalf of that ownership under cryptographic, time-boxed capability grants. The central original contribution is the Ownership Thesis: the axiom that every layer of the stack must be reducible to a claim controlled by a decentralized identifier the subject holds directly, not one that is issued or held by an operator on the subject's behalf. From this axiom we derive a seven-layer reference architecture and formalize the Synthetic Web as the class of network activity structurally defined by agent-executed action under such capability grants. We position each layer against the relevant standards (W3C DID Core v1.1, W3C Verifiable Credentials Data Model v2.0, Model Context Protocol, Agent-to-Agent v1.0.0, NIST CAISI AI Agent Standards Initiative) and provide a security analysis covering key management, alias unlinkability via BBS+ selective disclosure, Sybil resistance, delegated-authority scope creep, and data-sovereignty enforcement, with explicit identification of open problems that require further verification.*

Index Terms*—Digital sovereignty, decentralized ownership, self-sovereign identity, Web4, decentralized identifiers, verifiable credentials, autonomous agents, Model Context Protocol, Agent-to-Agent protocol, BBS+ signatures, decentralized compute, decentralized name service, network architecture.*

Disclosure. *The reference architecture presented here is developed and operated by the LINKSPREED GROUP. A companion technical report describing the concrete component implementation is available separately. The present paper describes the architecture at the level of abstract layers and standards composition and does not depend on any specific vendor.*

I. INTRODUCTION

The evolution of the World Wide Web is conventionally narrated as a sequence of expanding verbs: Web1 was READ, Web2 added WRITE, and Web3 added OWN. Each transition relocated a specific form of power along the value chain — from publishers who held the servers, to platforms who held the identity graph and monetization, to ledger-anchored token holders who could hold assets without a platform's permission. Each transition was also incomplete. Web2 gave users the ability to write but not the ability to own what they had written; Web3 gave a small number of primitives — tokens, wallets, dApps — back to their holders, but left the surrounding infrastructure of identity, connectivity, compute, and naming under precisely the centralized ownership Web2 had entrenched.

This paper argues that the next transition is not another incremental verb bolted onto the previous stack; it is a change of architectural axiom. It is the assertion that sovereign ownership must be co-specified across every operational plane of the network stack — identity, physical connectivity, compute, naming, developer tooling, application delivery, and hardware — rather than added as a policy overlay to one layer of it. We call this the Ownership Thesis, and we formalize it in Section II as the architectural axiom from which the rest of the specification is derived. The thesis is orthogonal to the specific standards it composes; it is the specific claim this paper defends against every previous generation of the web.

The immediate technical motivation is that autonomous software agents — first-class network participants that retrieve data, invoke tools, and take multi-step action on behalf of a user — make the ownership question urgent rather than philosophical.

In February 2026 the U.S. National Institute of Standards and Technology's Center for AI Standards and Innovation (NIST CAISI) launched an AI Agent Standards Initiative organized around the observation that most agent-to-API authentication still resolves to a static credential scoped to the human developer, not to the acting agent [6]. That gap is not solvable at the identity layer alone; it requires a stack in which every layer above the identity substrate consumes the same sovereign primitive — which is precisely what the Ownership Thesis prescribes.

This paper formally establishes Web4 as a protocol-level paradigm for the next generation of the internet, characterized by the systematic extension of sovereign ownership across every operational plane of the network to enable secure, autonomous agentic action (READ + WRITE + OWN + ACT). The remainder of the paper is organized as follows: Section II states and defends the Ownership Thesis and situates it against Web1, Web2, Web3, and against competing contemporary definitions of Web4. Section III specifies the seven-layer architecture that realizes it. Section IV specifies each layer in depth. Section V provides a security and identity analysis, and Section VI concludes.

II. THE OWNERSHIP THESIS

A. The Central Claim

The Ownership Thesis, in its shortest form, is the following axiom: every layer of the stack must be reducible to a claim controlled by a decentralized identifier that the subject holds directly, not one that is issued or held by an operator on the subject's behalf. This is not a slogan about user experience; it is a testable architectural predicate. For any layer L in the stack and any resource R visible at L, there must exist a path from R to a subject-controlled decentralized identifier such that the subject can, unilaterally, revoke R's association with themselves. Where such a path does not exist, the layer is not sovereign; it is rented.

The thesis is the original contribution of this paper. The individual technical components it composes — Decentralized Identifiers [1], [2], Verifiable Credentials [3], the Model Context Protocol [4], the Agent-to-Agent protocol [5], and the BBS+ signature scheme [7], [8] — exist independently. What has not been done before, and what this paper defends, is the systematic derivation of a full network stack — identity, physical connectivity, compute, naming, tooling, applications, and hardware — from the single axiom above.

B. From Read to Own to Act — The Evolution of the Web

Table I situates the Ownership Thesis in the historical evolution of the web. Web1, the read-only era of the 1990s, gave users access to static hypertext but no ability to alter it; whoever owned the server owned the page. Ownership was implicit and monopolar, but the network was fundamentally horizontal — a personal homepage was as reachable as a corporate one, and the identity of the reader was not itself a monetizable object.

Web2, the read-write era beginning around 2004, gave users the ability to produce content but silently collapsed the ownership question. Users produced posts, photos, videos, and social graphs; platforms captured the identity, the graph, the data, and the monetization. The industrial reality behind two decades of "user data" complaints is not an accident of business model but a consequence of layer choice: identity, connectivity, compute, and naming were all specified independently and outsourced to different operators, and the user therefore had no single point at which to assert ownership. Each layer was someone else's contract.

Web3, from approximately 2017 onward, reintroduced the notion of ownership at the ledger level: an asset could be held cryptographically, without a platform's permission, and a transaction could be executed against a smart contract without an intermediary's approval. This was an unambiguous advance, but it was confined to a narrow slice of the stack. Tokens and wallets became sovereign; identity, connectivity, compute, naming, applications, and hardware remained rented from the same operators that had governed them under Web2. A user with a self-custodied wallet was still reachable only through a mobile network operator's subscriber identifier, still resolvable only through the ICANN name hierarchy, still hosted only on hyperscale cloud infrastructure, and still identifiable only through a platform-issued profile.

Web4, as specified in this paper, extends the ownership primitive Web3 introduced at the ledger layer to the entire operational surface of the network. It also introduces the fourth verb: ACT. Where Web1 was READ, Web2 was READ+WRITE, and Web3 was READ+WRITE+OWN, Web4 is READ+WRITE+OWN+ACT — because an owner's autonomous agents must be

able to act on their behalf, under cryptographic capability grants tied to the owner's decentralized identifier rather than to a developer-scoped API key.

Table I is not a replacement schema; it is a cumulative superset, with each earlier verb preserved and a new ownership surface added at each transition. Web4 does not deprecate reading, writing, or ledger-anchored ownership — it composes them with the missing sovereignty guarantees at the layers Web1–Web3 left rented.

TABLE I

Web1 — Web2 — Web3 — Web4: The Ownership Progression

Era	Web1 (1990s)	Web2 (2004+)	Web3 (2017+)	Web4 (2026+)
Verb	READ	READ + WRITE	READ + WRITE + OWN	**READ + WRITE + OWN + ACT**
Paradigm	Library	Platform	Ledger	**Sovereign Stack**
Identity	Server-owner	Platform-issued profile	Wallet address	**Sovereign DID + Aliases**
Ownership scope	Publisher of the page	Platform of the graph	Tokens and ledger records	**Identity, connectivity, compute, naming, tooling, apps, hardware**
Failure mode	No participation	Digital dispossession	Sovereign tokens on rented infrastructure	**Open problems in cryptographic audit (see Sec. V)**

C. Digital Dispossession — The Problem Web4 Solves

Between roughly 2005 and 2025, the dominant user experience of the internet was defined by what we term digital dispossession: a state in which the user produces the value — attention, content, relationships, transactions — while an operator captures the ownership. Digital dispossession is not a moral judgment; it is an architectural observation. It is what necessarily follows when the identity layer, the naming layer, the compute layer, and the connectivity layer are each held by different centralized operators who have no shared incentive to expose an ownership primitive back to the user. Web4's original claim is that ownership must therefore be co-specified across all seven layers, or it is not ownership. A sovereign identity without a sovereign name is unreachable; a sovereign name without sovereign compute has nowhere to run; sovereign compute without sovereign connectivity cannot be reached at all. Section IV shows that each of the seven Web4 layers is a direct architectural response to one of these gaps.

D. The Sovereignty Principle as an Architectural Axiom

We restate the Ownership Thesis as a formal axiom and enumerate its direct consequences at each layer. Let a subject S be identified by a sovereign decentralized identifier DID(S). Then for every resource R at every layer $L \in \{L1, L2, L3, L4, L5, L6, L7\}$:

(i) R must carry a controller field resolvable to DID(S);
(ii) S must be able to produce a zero-knowledge attestation of control over R without disclosing the underlying key material;
(iii) any capability delegated over R — for example to an autonomous agent — must be expressed as a signed, time-boxed grant referencing DID(S) as issuer; and
(iv) revocation of that grant must propagate to every layer at which R is visible.

These four consequences are what Sections IV-A through IV-G specify concretely for identity, connectivity, compute, naming, tooling, applications, and hardware respectively.

E. What Web4 Adds Beyond the Standards

The Ownership Thesis is not implied by any of the standards Web4 composes; it must be added on top of them. The W3C Decentralized Identifiers specification defines the identifier and its resolution model [1], [2] but is deliberately silent on what happens at the six higher layers of a real deployment; the Verifiable Credentials data model [3] defines the credential envelope but not what layers of the network consume it; the Model Context Protocol [4] defines an agent's tool interface but not the identity under which it is invoked; the Agent-to-Agent protocol [5] defines peer-to-peer agent coordination but not the sovereignty of the delegating principal; and the NIST CAISI initiative [6] correctly identifies the agent-authentication gap but does not itself specify a stack that closes it. Web4's original contribution is the composition: a specification in which every one of these standards is bound to the same sovereign DID, so that ownership at Layer 1 is inherited unbroken by every layer above it.

F. Related and Competing Definitions of Web4

The term "Web 4.0" and "Web4" have been used in the literature and in policy documents with several distinct meanings, and a rigorous specification requires disambiguating these before proceeding.

First, the European Commission's 2023 Communication COM(2023) 442 [14] adopts an immersive definition, characterizing Web 4.0 as the fourth generation of the World Wide Web in which "advanced artificial and ambient intelligence, the Internet of Things, trusted blockchain transactions, virtual worlds and XR capabilities" combine to fully integrate digital and physical environments. This definition emphasizes the interface modality (immersive, spatial, XR-based) rather than the ownership properties of the underlying stack.

Second, a symbiotic definition [15] — occasionally called the Symbiotic Web or the Intelligent Web — foregrounds a reciprocal relationship between human cognition and machine execution in which autonomous systems act proactively on the user's behalf. This definition is also primarily about interaction modality and machine intelligence rather than about the sovereignty of the substrate.

Third, a recent agentic definition [16] frames Web4 as the network in which AI agents become first-class participants that browse, transact, and negotiate independently, focusing on the enablement of machine autonomy at the application layer.

The present paper is orthogonal to, and composable with, all three of these framings. It is a substrate-level rather than an interface-level or interaction-level specification: the immersive, symbiotic, and agentic characteristics identified in [14], [15], and [16] can each be implemented on top of a sovereign-ownership substrate of the kind formalized here, but not the reverse. Where necessary in the remainder of this paper, we refer to our position as the sovereign-ownership definition of Web4 to distinguish it explicitly from the three prior framings above.

III. SYSTEM ARCHITECTURE

A. Layering Rationale

The seven-layer architecture is the direct engineering realization of the Ownership Thesis. Rather than each service maintaining its own account system, every principal in the network — human, institution, hardware device, or software agent — is issued a single sovereign Core Identifier from which context-scoped, independently revocable Alias Identifiers descend. A second design premise is that physical connectivity, compute, and naming must be part of the same architectural specification as the applications that consume them, so that a subject's sovereignty at Layer 1 is not undone by dependencies on centrally operated services at Layers 2, 3, or 4.

B. Seven-Layer Overview

Table II summarizes the seven layers. Layers L1–L4 form the core protocol path analyzed in depth in this paper: sovereign identity substrate, sovereign physical access layer, decentralized compute mesh, and intelligent routing and naming. Layers L5–L7 are the application-facing planes — DID-authenticated developer tooling and services, sovereign super-application delivery, and certified consumer hardware. Throughout the remainder of the paper we refer to these layers exclusively by their functional role. A concrete component-level realization is presented in a companion technical report [10].

TABLE II

The Web4 Seven-Layer Reference Stack

Layer	Designation	Functional Role
L1	**Sovereign Identity Substrate**	DID method anchoring all higher-layer identity references; Core Identifier and context-scoped Alias Identifiers.
L2	**Sovereign Physical Access Layer**	Carrier-independent mobile connectivity (eSIM); DID-native voice and messaging endpoints.
L3	**Decentralized Compute Mesh**	Agentic-compute mesh; zero-knowledge sharded storage; Byzantine-fault-tolerant consensus.
L4	**Intelligent Routing and Naming Layer**	Decentralized name service replacing the ICANN/DNS hierarchy inside the Web4 stack; DID-controlled name records.
L5	**DID-Authenticated Service Layer**	Sovereign SaaS layer; developer tooling; user state written to DID-addressable key-value storage.
L6	**Sovereign Application Delivery Layer**	Super-application delivery plane; tenant-isolated deployment with shared identity substrate.
L7	**Sovereign Hardware Layer**	Consumer hardware operating as personal server; hardware-bound key custody; local Web4 execution.

C. Request Lifecycle

A request across the seven-layer stack proceeds as follows. A client device — typically an L7 endpoint reaching the network over an L2 sovereign connection — issues a DID-signed request to the receiving L6 instance. L6 dispatches the underlying intent through L5, which resolves the target endpoint via the L4 decentralized name service and consults L1 for a scoped capability grant tied to the acting Alias Identifier. The resulting action is executed against the L3 compute mesh and anchored as an independently addressable artifact under L4's namespace, so that both pre-action authorization and post-action attribution are cryptographically anchored rather than log-based.

IV. LAYER SPECIFICATION

A. L1 — Sovereign Identity Substrate

Layer 1 is the sovereign identity substrate: a Decentralized Identifier (DID) method in the sense defined by the W3C DID Core specification — a globally unique, URI-formatted identifier which resolves to a DID Document containing verification methods, service endpoints, and controller information [1], [2]. Resolution proceeds through the Layer 4 decentralized name service, which serves the verifiable-data-registry role defined in DID Core [2].

The substrate defines a two-tier identity model. The Core Identifier is the immutable, sovereign root of a subject's identity; it is never disclosed directly to third-party services and is used exclusively for first-party operations. Alias Identifiers are subordinate, context-scoped identifiers — for example a financial alias, a professional alias, or a social alias — issued as Verifiable Credentials that cryptographically reference the Core Identifier without disclosing it to a relying party [3]. Authentication follows a passwordless flow combining WebAuthn with zero-knowledge proofs, and access is bounded by granular scopes; mandatory Multi-Factor Authentication protects critical operations, and transparent audit-log endpoints record every issuance and verification event.

The substrate is intentionally decoupled from the physical device on which it is used: biometric material for enrollment is stored locally on the Layer 7 endpoint rather than on a remote server, and proofs to relying parties are produced as zero-knowledge attestations rather than by transmitting the underlying attributes. This decoupling is what allows the same DID to be presented across the six higher layers without any of those layers gaining custody of the underlying key material.

The substrate is extended in current work from human-only identity to a comprehensive multi-entity framework covering six specialized entity types: Human Individual, Organization, AI Agent, Hardware Device, Role, and Task, each with a tailored enrollment flow. Autonomous agents register their own DID accounts through a machine-readable capability protocol that

advertises operational capabilities and enables automated handshakes with other nodes. Static secrets and long-lived access keys are replaced with dynamic, single-use or multi-use provisioning tokens issued from a controlling human or organization principal, from which the principal can monitor fleet activity, resource consumption, and trust scores in real time.

B. L2 — Sovereign Physical Access Layer

Layer 2 is the physical mobile-access layer of the stack. It provides global eSIM connectivity independent of a single national mobile network operator, and exposes DID-native primitives for voice and messaging so that a subject's decentralized identifier can act as the addressable endpoint for a call or a chat message in the same way a phone number does under the E.164 numbering plan, without the number itself ever being issued or held by a telecommunications carrier. Layer 2 is the layer at which the physical connection to the network is bound to the identity substrate rather than to an operator-issued subscriber identifier, so that a subject retains reachability under their own DID when moving between jurisdictions or carriers.

C. L3 — Decentralized Compute Mesh

Layer 3 is the decentralized compute layer of the stack. It composes idle end-user hardware into an agentic-compute mesh in which state is synchronized under a Byzantine-fault-tolerant consensus scheme and persisted through a zero-knowledge storage layer that shards ciphertext across participating nodes rather than concentrating it in a single operator's cluster. Layer 3 nodes present a common execution environment that exposes the identity substrate at the operating-system level so that a call into a compute resource is authorized under the same DID and scope as a call into any other Web4 service.

The architectural consequence is that compute is procured under the same sovereignty model as identity: the workload is bound to the DID that authorized it and to the capability grant under which it runs, rather than to an account issued by the compute operator. This is the layer at which the Synthetic Web (Section IV-H) is executed; an autonomous agent acting under a DID-derived, time-boxed capability grant consumes compute cycles on Layer 3 nodes and records its outputs on the same mesh, without needing an intermediate hyperscale-provider account.

D. L4 — Intelligent Routing and Naming Layer

Layer 4 is the routing and naming plane. It provides two functions that are conventionally furnished by separate operators on the legacy internet: intelligent network routing between Web4 endpoints, and a decentralized name service that replaces the ICANN/DNS hierarchy for names inside the Web4 stack. Under this layer, a name is registered as a record whose controller is a decentralized identifier, so ownership of the name is bound to sovereign identity rather than to a recurring registrar contract.

Layer 4 additionally provides the operating-system integration surface — including a Web4-aware browser and search interface — at which higher-layer applications (Layers L5–L7) resolve name records into Layer 3 endpoints. Because Layer 4 is the verifiable-data-registry the Layer 1 DID method depends on for DID Document resolution, its consistency and update-propagation guarantees directly bound the security properties analyzed in Section V [2].

E. L5 — DID-Authenticated Service Layer

Layer 5 is the developer-facing service layer of the stack, providing DID-authenticated tooling that consumes the identity, routing, and compute layers below it without reimplementing them. Layer 5 is organized into two suites: a business-to-business suite oriented at enterprise deployments, and a business-to-consumer suite of applications supported by a privacy-friendly, non-tracking model. Layer 5 applications do not maintain independent user databases; user state is written to a DID-addressable key-value storage that is exposed to authenticated applications as an extension of the Core Identifier, so that a user's data associated with a Layer 5 application remains addressable under the same DID after the application itself is retired or replaced.

F. L6 — Sovereign Application Delivery Layer

Layer 6 is the application-delivery plane. It provides Super Applications as a Service: a customer can deploy a fully-featured social network or super-application on top of the underlying stack, in which the desired network is described in natural language and the corresponding Layer 6 configuration is generated and deployed against the lower layers. Layer 6 instances are tenant-isolated so that operators can deploy independently branded, independently configured super-applications while still relying on the shared identity substrate, and a credential issued in one instance is portable to another without

re-registration. Because Layer 6 is the layer at which both human-session traffic and Synthetic-Web traffic terminate, it is the layer at which the distinction analyzed in Section V becomes operationally visible.

G. L7 — Sovereign Hardware Layer

Layer 7 is the consumer-hardware layer of the stack. Its two flagship components are a personal-server smartphone that hosts the user's Layer 6 tenant and Layer 1 key material on-device rather than in a remote provider's cluster, and a low-power household appliance that brings a Layer 3 mesh node into the home so that a household's Web4 activity can execute locally rather than being routed to third-party infrastructure. Layer 7 is therefore the layer at which the sovereignty guarantees asserted at Layers L1–L4 are physically realized on a device the user actually controls; where a deployment requires hardware-bound key custody, Layer 7 is the layer at which that root of trust is provisioned.

H. The Synthetic Web

The Synthetic Web is defined here as the subset of Web4 network activity that is originated, negotiated, or executed by an autonomous agent acting under a DID-derived, time-boxed capability grant, rather than in direct, synchronous response to a live human input. It is a structural rather than a content-based definition: traffic is classified by the authorization and execution model under which it was produced, not by whether the resulting artifact is text, a financial transaction, or a generated credential.

The execution model at Layer 3 and the routing model at Layer 4 together compose two agent-interoperability patterns currently standardized separately. Agent-to-tool interaction — an agent invoking a callable function exposed by a server, together with contextual resources and structured input/output — follows the Model Context Protocol, a JSON-RPC 2.0 host/client/server topology whose servers expose Tools, Resources, and Prompts and whose current transport binding is Streamable HTTP with OAuth 2.1 authorization [4]. Agent-to-agent interaction — an agent delegating a task to, or requesting a capability from, an independently operated agent — follows the Agent-to-Agent (A2A) protocol, which represents agent capabilities via signed Agent Cards and supports synchronous request/response, streaming updates via Server-Sent Events, and asynchronous push notifications for long-running, human-in-the-loop tasks [5].

Web4 does not introduce a competing protocol at either layer; it constrains both to operate under DID-derived, capability-scoped authorization, so that every tool invocation and every delegated task is authorized under a time-boxed, narrowly scoped capability grant tied to a specific Alias Identifier rather than to a long-lived, developer-scoped API key — directly closing the gap NIST CAISI identifies as the current default failure mode of agent-to-API authentication [6]. This is the ACT verb of the Ownership Thesis in its concrete form. A Web4 capability grant is a signed structure containing the issuing principal DID, the acting subject DID, the target audience DID, the scope, a not-before timestamp, an expiry timestamp, and a unique grant identifier.

Synthetic-Web traffic is thereby characterized by three structural properties that distinguish it from the conventional request/response model of the human-session web: (i) delegated, scoped authorization — an agent acts under a capability token bound to a specific Alias Identifier and a specific, time-boxed grant, never under the subject's Core Identifier; (ii) standing-intent execution — an agent may act continuously against a previously registered intent rather than only within the lifetime of an interactive session; and (iii) machine-originated artifacts as first-class resources — data, documents, and transactions produced autonomously by an agent are registered under Layer 4 as independently addressable resources, not as transient scratch state internal to a single request.

V. SECURITY AND IDENTITY ANALYSIS

A. Cryptographic Foundations

The DID Core specification is deliberately agnostic to signature scheme and key-agreement algorithm, leaving those choices to the method specification [1], [2]. The reference implementation of the Layer 1 substrate uses Ed25519 for authentication and assertion signatures and X25519 for key agreement, consistent with common practice among deployed DID methods. A formal method specification — analogous to the specifications collected in the W3C DID Specification Registries [12] — would need to define supported verification-method types explicitly rather than leaving them to reference-implementation convention; publishing that specification is identified in Section VI as a concrete open task.

B. Alias Unlinkability and Selective Disclosure

The claim that an Alias Identifier is unlinkable to its Core Identifier, or to a subject's other aliases, does not follow automatically from the fact that each alias is issued as a separate credential. If alias issuance and verification both route through the same Layer 4 registry, and that registry logs issuance and verification events, an operator or a verifier colluding with the registry can in principle correlate aliases by timing, request pattern, or issuance metadata even without ever seeing the Core Identifier in plaintext.

Meaningful unlinkability requires selective-disclosure cryptography. The BBS+ signature scheme, currently proceeding through IRTF CFRG standardization, is a pairing-based multi-message digital signature — instantiated over the BLS12-381 curve — that allows a holder to derive a zero-knowledge proof of possession of a valid signature while disclosing only a chosen subset of the signed messages [7], [8]. Two properties are structurally relevant to Alias Identifiers: (i) selective disclosure of individual attributes without revealing the remainder of the credential or the underlying signature, and (ii) unlinkability, meaning that two proofs derived from the same credential cannot be cryptographically correlated by a verifier [7], [8], [13]. The architecture as specified anticipates BBS+ (or a comparable selective-disclosure scheme) at the verification point of an Alias Identifier, but a claim that alias unlinkability holds in a given deployment should be treated as contingent on an explicit implementation of the scheme and independent cryptographic audit, rather than as a property guaranteed by the DID-based architecture alone.

C. Threat Model

Four risk categories are salient. First, key compromise and recovery: as in any self-sovereign identity scheme, loss or compromise of a subject's Core Identifier key material is catastrophic by default, and mitigating it requires a recovery mechanism — social recovery via designated guardians, or hardware-backed custody via Layer 7 — that is itself a trust assumption external to the cryptography. Second, revocation latency: an Alias Identifier or capability grant that has been revoked in the Layer 4 registry is only as effective as the propagation delay before relying parties observe the revocation; a request authorized in the window between compromise and propagated revocation succeeds despite the revocation. Third, delegated-authority scope creep: Web4 capability grants must be narrowly scoped and time-boxed by construction, since an agent holding an over-broad or long-lived grant reintroduces exactly the risk class NIST CAISI identifies in current static-API-key practice — uncontrolled autonomy and an expanded attack surface from agent-held credentials [6]. Fourth, Sybil resistance: the DID layer alone does not prevent an actor from generating an arbitrary number of identifiers, since keypair generation is unpermissioned by design; Sybil resistance at the population level — for example, one identifier per natural person — requires binding issuance to an external identity-proofing process such as biometric enrollment or government-credential verification, which is a policy layer external to the cryptographic protocol.

D. Comparative Positioning

Relative to fully permissionless, ledger-anchored DID methods, the specified resolution model trades decentralization purity for throughput, cost, and governance simplicity: the Layer 4 name service is an operated verifiable-data-registry rather than a permissionless ledger, which means resolution does not depend on blockchain consensus latency or transaction fees, but also means the operator is a trust point that a fully ledger-anchored method would not introduce. Relative to the W3C Verifiable Credentials Data Model [3], the Alias Identifier system is a direct application of the VC model to the specific problem of context-scoped, per-relying-party identity presentation, rather than a departure from it. Relative to MCP and A2A, Web4 does not compete with either protocol; it constrains both to operate under identity- and capability-scoped authorization rather than the ambient, developer-scoped authorization common in current agent deployments [4], [5].

E. Alignment with NIST CAISI

The CAISI AI Agent Standards Initiative, announced 17 February 2026, organizes its work along three pillars: facilitating industry-led development of agent standards, fostering community-led open-source protocol development, and advancing research in agent security and identity [6]. In parallel, the National Cybersecurity Center of Excellence published a February 2026 concept paper on AI agent identity and authorization, identifying prompt injection and accountability gaps in autonomous action chains as leading security concerns [13]. The composition specified in Section IV-A, combined with the execution model in Section IV-C and the naming anchoring in Section IV-D, is a concrete architectural response to those gaps: every autonomous action is executed under a signed capability grant that names the acting agent's DID, the invoking principal, the target tool or agent, the specific scope, and a mandatory expiry, so that both pre-action authorization and post-action attribution are cryptographically anchored rather than log-based.

F. Data Sovereignty Enforcement

Data sovereignty in Web4 is enforced across three layers rather than at a single point. At Layer 1, sensitive attributes remain in the subject's custody on their Layer 7 device and are disclosed to relying parties only as zero-knowledge attestations. At Layer 2, the physical connection to the network is bound to the DID rather than to an operator-issued subscriber identifier, so that a subject's reachability is not conditioned on a single carrier's continuing consent. At Layers 3–4, workloads and namespace records execute against a mesh of independently operated nodes rather than in a single operator's cluster. Combined, these controls are intended to provide the practical guarantee that data-residency regimes such as the EU General Data Protection Regulation aim at — that data about a subject in a given jurisdiction remains subject to, and auditable under, that jurisdiction's law, independent of where any single operator is headquartered [11].

VI. CONCLUSION

This paper has introduced and defended the Ownership Thesis — the axiom that every layer of the network stack must be reducible to a claim a sovereign decentralized-identifier subject holds directly — and has derived from it a seven-layer reference architecture that integrates decentralized identity, physical mobile access, decentralized compute, intelligent routing and naming, DID-authenticated developer tooling, super-application delivery, and certified consumer hardware into a single stack. It has formalized the Synthetic Web as the class of network activity structurally defined by delegated, capability-scoped, agent-executed authorization rather than by content type. Each core layer has been positioned against the relevant existing standardization efforts — the W3C DID Core v1.1 Candidate Recommendation of March 2026, the W3C Verifiable Credentials Data Model v2.0, the Model Context Protocol, the Agent-to-Agent protocol v1.0.0, and the NIST CAISI AI Agent Standards Initiative of February 2026 — on the premise that Web4's contribution is the composition of those standards into a stack that satisfies the Ownership Thesis, not any single standard on its own.

We have further disambiguated the sovereign-ownership definition of Web4 defended here from three prior contemporary framings of the term: the immersive-interface definition adopted by the European Commission in COM(2023) 442 [14], the symbiotic definition of the Symbiotic Web [15], and the agentic-autonomy definition popularized in early 2026 [16]. The sovereign-ownership definition is a substrate-level rather than an interface-level or interaction-level specification and is composable with each of the prior three framings above.

The security analysis in Section V identifies concrete open problems rather than resolved guarantees. Alias unlinkability is contingent on an explicit BBS+ (or comparable selective-disclosure) implementation and independent cryptographic audit. Population-level Sybil resistance depends on an identity-proofing process that must be specified as policy, external to the cryptographic protocol. A formal DID method specification, aligned with the entries in the W3C DID Specification Registries, must still be published for independent implementation and conformance testing. We take the position that these are the correct next steps for this architecture, and that the value of the specification given here is in making the Ownership Thesis — the specific claim that ownership must be co-specified across every layer — precise and falsifiable, rather than in asserting that any single deployment already fully satisfies it.

The reference implementation from which this architecture has been abstracted is operated by the LINKSPREED GROUP; a companion technical report describing that concrete implementation is available separately [10].

ACKNOWLEDGMENT

The author thanks the LINKSPREED GROUP Web4 Division for supporting the development of the reference implementation from which this architecture has been abstracted.

REFERENCES

[1] W3C, "Decentralized Identifiers (DIDs) v1.0," W3C Recommendation, 2022. [Online]. Available: https://www.w3.org/TR/did-1.0/

[2] W3C, "Decentralized Identifiers (DIDs) v1.1," W3C Candidate Recommendation Snapshot, 5 Mar. 2026. [Online]. Available: https://www.w3.org/TR/2026/CR-did-1.1-20260305/

[3] W3C, "Verifiable Credentials Data Model v2.0," W3C Recommendation. [Online]. Available: https://www.w3.org/TR/vc-data-model-2.0/

[4] Model Context Protocol Working Group, "Model Context Protocol Specification," 2024–2026. [Online]. Available: https://modelcontextprotocol.io/specification

[5] A2A Project, "Agent2Agent (A2A) Protocol Specification, v1.0.0," 2026. [Online]. Available: https://a2aprotocol.org

[6] NIST Center for AI Standards and Innovation (CAISI), "AI Agent Standards Initiative," announcement, 17 Feb. 2026. [Online]. Available: https://www.nist.gov/artificial-intelligence/ai-agent-standards-initiative

[7] T. Looker, V. Kalos, A. Whitehead, and M. Lodder, "The BBS Signature Scheme," IRTF CFRG Internet-Draft draft-irtf-cfrg-bbs-signatures, Jul. 2025.

[8] S. Tessaro and C. Zhu, "Revisiting BBS Signatures," in Advances in Cryptology — EUROCRYPT 2023; IACR ePrint 2023/275.

[9] M. Herdina, "Sovereign Machine Identity: Autonomous Agent Enrollment, Agent Factory, and Zero-Trust Provisioning," Technical Report, LINKSPREED GROUP, Jun. 2026.

[10] M. Herdina, "Web4 Reference Implementation — A Seven-Layer Sovereign Stack: Companion Technical Report," LINKSPREED GROUP, 2026.

[11] European Parliament and Council, "Regulation (EU) 2016/679 (General Data Protection Regulation)," Official Journal of the European Union, 2016.

[12] W3C DID Working Group, "DID Specification Registries," W3C Group Note.

[13] Cloud Security Alliance AI Safety Initiative, "NIST CAISI: AI Agent Standards and the Enterprise Compliance Imperative," Research Note, Mar. 2026.

[14] European Commission, "An EU initiative on Web 4.0 and virtual worlds: a head start in the next technological transition," Communication COM(2023) 442 final, 11 Jul. 2023.

[15] S. Aghaei, M. A. Nematbakhsh, and H. K. Farsani, "Evolution of the World Wide Web: From Web 1.0 to Web 4.0," International Journal of Web & Semantic Technology, vol. 3, no. 1, pp. 1–10, 2012.

[16] S. Wen, "The Web4 Manifesto: Toward an Agentic Internet," independent publication, Feb. 2026.